

## REMARKS

Claims 1-35 were pending as of the action mailed on April 9, 2007. Claims 1, 15, and 29 are in independent form.

Claims 1-4, 6-8, 14-18, 20-22, and 28 are being amended. Claims 9-10, 23-24, and 29-35 are being cancelled. Claims 36-38 are new. No new matter has been added. Support for the amendments can be found, for example, from page 6, line 14, through page 7, line 19, of the specification.

Reexamination of the application and reconsideration of the action are respectfully requested in light of the foregoing amendments and the following remarks.

### **Rejection under Section 101**

Claims 15-35 were rejected as allegedly directed to non-statutory subject matter.

Claims 15-28. Claim 15 is an independent claim. Claims 16-22 and 25-28 all depend from claim 15. Claim 15 recites, in pertinent part,

15. A computer system comprising:  
a common source unit . . . ; and  
a means for producing . . . ; and  
a means for loading . . . .

The claim clearly recites a practical application and tangible result, namely loading executable test code and the executable production code in a development runtime environment in that executes the executable test code and performs the tests directed by the executable test code.

Moreover, the claim includes limitations to which the sixth paragraph of 35 U.S.C. § 112 apply. “Where means plus function language is used to define the characteristics of a machine or manufacture invention, such language must be interpreted to read on only the structures or materials disclosed in the specification and “equivalents thereof” that correspond to the recited function. Two *en banc* decisions of the Federal Circuit have made clear that the USPTO is to interpret means plus function language according to 35 U.S.C. § 112, sixth paragraph.” MPEP §

2106 page 2100-7 (Rev. 5, Aug. 2006) (emphasis added) Consequently, the claim is directed to statutory structure and subject matter.

For the foregoing reasons, the applicant respectfully submits that the rejection of claims 15-29 under § 101 should be withdrawn.

Claims 23-24 and 29-35. The rejection as to claims 23-24 and 29-35 is moot, as those claims have been cancelled.

### **Rejections under Section 103**

Claims 1-35 were rejected as allegedly unpatentable over Borland C++ 3.0 (“C++”) in view of McConnell, Code Complete (“CC”).

Claim 1. In rejecting claim 1, the examiner stated that the claims “read on well known conditional compilation . . . .”

The applicant respectfully disagrees.

Claim 1 has been amended to recite:

loading the executable test code with the executable production code in a development runtime environment in response to a setting of a global switch, wherein the development runtime environment executes the executable test code and performs the tests directed by the executable test code.

In conditional compilation, the test code is either compiled or not compiled. The claim, on the other hand, recites a switch that specifies whether test code is loaded into the production environment, not whether the code is compiled. More particularly, the recited switch is a global switch that governs whether any test code is loaded into the product environment.

In addition, the claim now recites that the test and production code are loaded in a development environment that performs the tests directed by the test code.

The applicant respectfully submits that these features are not shown by C++ or CC, alone or in combination.

For the foregoing reasons, the rejection of claim 1 should be withdrawn.

Claim 2. Claim 2 depends from claim 1 and is allowable for at least that reason.

Claim 2 has been amended to recite that:

test source code and the production source code are written in a common programming language, and the test source code is identified by a key word that is recognized as a language extension to the common programming language.

CC teaches a preprocessor (see page 98), not a language extension or a key word identifying a source code statement as being test rather than production code.

For the foregoing additional reason, the rejection should be withdrawn.

Claim 3. Claim 3 depends from claim 1 and is allowable for at least that reason.

Claim 3 has been amended to recite that:

the executable test code includes test methods that can be called from test tools of the development runtime environment.

The applicant respectfully submits that this feature is not shown by C++ or CC, alone or in combination.

For the foregoing additional reason, the rejection of claim 3 should be withdrawn.

Claim 4. Claim 4 depends from claim 1 and is allowable for at least that reason.

Claim 4 has been amended to recite that:

the executable test code includes test methods that test production methods of the executable production code by calling production methods of the executable production code

Thus, the unit testing of the common source code unit is performed by test code that is in the common source code unit, and not external to it. (See, e.g., application, page 1, lines 10-18, and page 3, lines 4-8) The applicant respectfully submits that this feature is not shown by C++ or CC, alone or in combination.

For the foregoing additional reason, the rejection of claim 4 should be withdrawn.

Claim 6. Claim 6 depends from claim 1 and is allowable for at least that reason.

Furthermore, claim 6 recites “checking static references from the production source code to the test source code” where both are included in a “common source code unit” (recited in claim 1).

The examiner states that this feature is taught by C++ on pages 22-24.

The applicant respectfully disagrees.

The cited pages describe features of the Borland command-line MAKE, which is used to keep executable versions of programs current, for programs that consist of many source files. C++, page 9. Claim 6 relates to a single “common source code unit”.

In addition, claim 6 has been amended to make clear that the checking in question is done to confirm that there are no static references from the production source code to the test source code. MAKE relates to very different kinds of processes that involve selecting files by name and processing them. C++, page 9.

For the foregoing additional reasons, the rejection of claim 6 should be withdrawn.

Claim 7. Claim 7 has limitations that parallel those of claim 6 and is allowable for corresponding reasons.

#### Remaining claims.

As the examiner noted, the remaining claims depend from or correspond to claims that have been addressed and are allowable for the corresponding reasons.

### **Conclusion**

For the foregoing reasons, the applicant submits that all the claims are in condition for allowance.

By responding in the foregoing remarks only to particular positions taken by the examiner, the applicant does not acquiesce with other positions that have not been explicitly addressed. In addition, the applicant's selecting some particular arguments for the patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist. Finally, the applicant's decision to amend or cancel any claim should not be

Applicant : Andreas Blumenthal, et al.  
Serial No. : 10/723,702  
Filed : November 26, 2003  
Page : 14 of 14

Attorney's Docket No.: 13913-152001 / 2003P00554  
US

understood as implying that the applicant agrees with any positions taken by the examiner with respect to that claim or other claims.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 26 Jun 07

/Hans R. Troesch/

Hans R. Troesch  
Reg. No. 36,950

**Customer No. 32864**

Telephone: (650) 839-5070  
Facsimile: (650) 839-5071

50422778.doc